高精度例题：

import java.util.Scanner;

import java.math.\*;

public class Main {

public static class point {

BigDecimal a, b;

}

public static point tempx = new point();

public static point find(point x, point y, point z) {

BigDecimal a, b, a1, a2, b1, b2, c1, c2, t, d;

t = BigDecimal.valueOf(2);

a1 = y.a.subtract(x.a);

b1 = y.b.subtract(x.b);

tempx.a = a1;

tempx.b = b1;

c1 = ((a1.multiply(a1)).add(b1.multiply(b1))).divide(t);

a2 = z.a.subtract(x.a);

b2 = z.b.subtract(x.a);

c2 = ((a2.multiply(a2)).add(b2.multiply(b2))).divide(t);

d = (a1.multiply(b2)).subtract(a2.multiply(b1));

tempx.a = x.a.add((c1.multiply(b2).subtract(c2.multiply(b1))).divide(d, 20, 0));

tempx.b = x.b.add((a1.multiply(c2).subtract(a2.multiply(c1))).divide(d, 20, 0));

return tempx;

}

public static BigDecimal distance(point x, point y)// 没有开根

{

BigDecimal temp;

temp = ((x.a.subtract(y.a)).multiply(x.a.subtract(y.a))).add((x.b.subtract(y.b)).multiply(x.b.subtract(y.b)));

return temp;

}

static point p[] = new point[5], temp = null;

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

int T, i;

T = s.nextInt();

for (i = 0; i < 4; i++)

p[i] = new point();

while (T-- > 0) {

for (i = 0; i < 4; i++) {

p[i].a = s.nextBigDecimal();

p[i].b = s.nextBigDecimal();

}

temp = find(p[0], p[1], p[2]);

if (distance(p[0], temp).compareTo(distance(p[3], temp)) < 0)

System.out.println("Accepted");

else

System.out.println("Rejected");

}

}

}

**大数总结**

BigInteger 可以用来解决数据的溢出问题。

下面我总结几种关于BigInteger的常用用法：

1。probablePrime和nextprobablePrime。(判断质数，并返回)

　　BigInteger.probablePrime(int x);

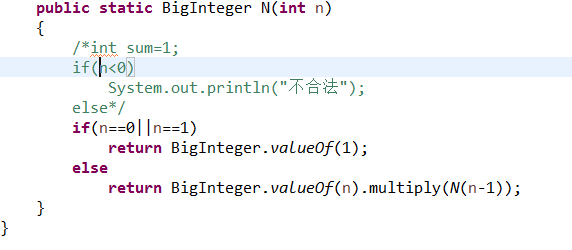
　　返回有可能是素数（质数），具有指定长度的正数BigInteger，返回可能是合数的概率不超过2的负100次方，

　　BigInteger.nextprobablePrime(int x)

　　返回大于此BigInteger的有可能是素数（质数），具有指定长度的正数BigInteger，返回可能是合数的概率不超过2的负100次方，

　2.valueOf()（对数据初始化）

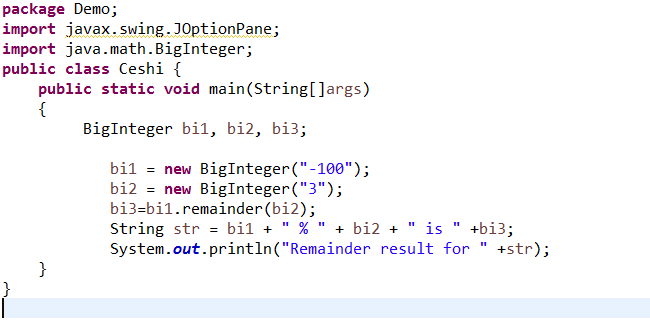
BigInteger valueOf(long val)

用法如下：

3四则运算（add（加）。subtract（减）。multiply（乘）。divide（除））

　　调用方式如上图，可以重复调用

4remainder（取余）

返回其值为取余后的BigInteger类型的值，例子如下

 调用的时候要保证数据是BigInteger类型的，可以用初始化（valueOf）或者new一个。

5divideAndRemainder（先除后取余，结果分别存在数组中）